Three New Algorithms for Projective Bundle Adjustment with Minimum Parameters

Adrien Bartoli, Peter Sturm

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Three New Algorithms for Projective Bundle Adjustment with Minimum Parameters

Adrien Bartoli — Peter Sturm

N° 4236

Août 2001

THÈME 3

ISSN 0249-6399 ISRN INRIA/RR--4236--FR+ENG

INRIA
RHÔNE-ALPES

Three New Algorithms for Projective Bundle Adjustment with Minimum Parameters

Adrien Bartoli , Peter Sturm

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet MOVI

Rapport de recherche n° 4236 — Août 2001 — 25 pages

Abstract: Bundle adjustment is a technique used to compute the maximum likelihood estimate of structure and motion from image feature correspondences. It practice, large non-linear systems have to be solved, most of the time using an iterative optimization process starting from a sub-optimal solution obtained by using linear methods. The behaviour, in terms of convergence, and the computational cost of this process depend on the parameterization used to represent the problem, i.e. of structure and motion.

In this paper, we address the problem of finding a minimal parameterization of projective structure and motion, i.e. when camera calibration is not available. Most of the existing parameterizations are either sub-optimal, in the sense that they do change the cost function, or quite complicated to implement, requiring different closed-form expressions, also called maps, to model every case. Without loss of generality, we restrict the problem to the minimal parameterization of the two-view projective motion, equivalent to the fundamental matrix. We propose to classify existing ways allowing to obtain a minimal set of parameters into three categories and present three new algorithms, one for each category. These algorithms are simple to implement and yield minimal parameterizations.

We also address the problem of minimally parameterizing homogeneous entities. We present what we call mapped coordinates, a tool allowing to optimize homogeneous entities over a minimal number of parameters. We make extensive use of this tool for both structure and motion parameterization.

We compare these algorithms with existing ones using both simulated data and real images. In the light of these experiments, it appears that the new algorithms perform better than existing ones in terms of computational cost while achieving equivalent performances in terms of convergence.

Key-words: Bundle Adjustment, Projective Space, Minimal Parameterization.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 Montbonnot-St-Martin (France)
Téléphone : +33 4 76 61 52 00 — Télécopie +33 4 76 61 52 52

Trois nouveaux algorithmes pour l'ajustement de faisceaux avec un nombre minimal de paramètres

Résumé : L'ajustement de faisceaux est une technique utilisée pour calculer l'estimation au maximum de vraisemblance d'une reconstruction à partir de correspondances de primitives images. En pratique, des systèmes non-linéaires conséquents doivent être résolus, la plupart du temps en utilisant une méthode d'optimisation itérative à partir d'une solution sous-optimale fournie par des méthodes linéaires. L'issue, en terme de convergence, et le coût de calcul de tels procédés dépend de la paramétrisation algébrique utilisée pour représenter le problème, c'est à dire la structure reconstruite et le mouvement entre les caméras.

Dans cet article, nous considérons le problème de trouver une paramétrisation minimale de la structure et du mouvement projectifs, c'est à dire lorsque les caméras ne sont pas calibrées. La plupart des paramétrisations existantes sont soit sous-optimales, dans le sens où elles changent la fonction de coût, soit compliquées à mettre en œuvre, plusieurs expressions analytiques, appellées aussi cartes, étant nécessaires pour modéliser tous les cas. Sans perte de généralité, nous restreignons le problème à celui de la paramètrisation minimale du mouvement entre deux vues dans le cas projectif, équivalent à la matrice fondamentale. Nous proposons de distinguer trois catégories de méthodes permettant d'obtenir un jeu de paramètres minimal et présentons trois nouveaux algorithmes, un par catégorie. Ces algorithmes sont simples à implémenter et permettent d'obtenir des paramétrisations minimales.

Nous considérons aussi le problème de la paramétrisation minimale des entités algébriques homogènes. Nous présentons les coordonnées carte, un outil permettant d'optimiser une entité homogène via un nombre minimal de paramètres. Nous utilisons fréquement cet outil dans les paramétrisations de la structure et du mouvement.

Nous comparons nos algorithmes avec d'autres existants en utilisant des données simulées et des images réelles. Avec ces expérimentations, il apparaît que les nouveaux algorithmes atteignent de meilleurs performances que ceux existants, en terme de coût, et des performances égales en terme de convergence.

Mots-clés : Ajustement de faisceaux, espace projectif, parametrisation minimale.

1 Introduction

Bundle adjustment is a standard photogrammetric technique for structure and motion optimization based on image feature correspondences across views [18, 22]. It has been demonstrated that bundle adjustment is the maximum likelihood estimator provided the error on measured features has gaussian distribution[1]. The interest of bundle adjustment has been experimentally confirmed since this enhances given estimates in most practical cases.

In this paper, we address the problem of bundle adjusting point features within a projective framework, which provides the flexibility of working with uncalibrated or partially calibrated views. Such a process is important since an accurate recovery of projective structure and motion is necessary to many other subsequent algorithms such as self-calibration.

It practice, bundle adjustment yields large non-linear optimization problems. Indeed, it consists in minimizing a cost function based on the discrepancy between reprojected features and actual ones, over the complete set of parameters of both structure and motion. Such a cost function has many local minima [14], which may cause the process to provide sub-optimal results. Moreover, despite the use of specific techniques, such as sparse Newton-type methods [18], bundle adjustment remains computationally expensive. Therefore, the key features of a bundle adjustment algorithm are its:

- properties of *convergence* to the global optimum;
- *computational cost* in terms of CPU time.

More precisely, bundle adjustment greatly depends on the algebraic parameterization of the problem, i.e. of structure and motion, and in particular, its numerical conditioning and whether the number of parameters employed is minimal.

The main problem one has to deal with is the *coordinate frame ambiguity*, in other words, choosing the projective basis where the reconstruction is expressed.

One of the first solutions that comes to mind consists in setting the coordinates of five reconstructed points so as to define the coordinate frame, but this implies to know a priori or to check that these points are in general position, so that they define a projective basis.

Another solution is to use a special form of camera matrices to define the coordinate frame. A minimum number of two views are necessary. In this paper and without loss of generality, we concentrate on the two-view case which is unfeasible only when the centers of projection of the two views are identical or the scene observed is planar[2]. The problem of expressing the reconstruction basis inside the two-view parameterization will be referred to as "minimally parameterizing the projective two-view motion". It has been shown that this is equivalent to minimally parameterizing the fundamental matrix [3, 7], problem that has no direct and global solution, in the sense that one can not define a unique mapping from a minimal set of 7 parameters to the entries of the fundamental matrix and which covers all two-view configurations.

[1] Note that this is true if the cost function used in the bundle adjustment is the sum of squared euclidean distances between estimated and actual image points, see §2.3.

[2] Note that this is unfeasible unless two or more points off the plane are not considered.

However, [2, 23] have proposed solutions to the problem of bundle adjustment with a minimal number of parameters. In this paper, we propose to classify these solutions into two categories and present a third one that is novel. We give new approaches and three new algorithms solving the problem and simple to use in practice.

Another important issue considered in this paper is the minimal parameterization of homogeneous entities, such as the projective structure. This problem, referred to as "minimally parameterizing homogeneous entities" is solved via what we call *mapped coordinates*. We make extensive use of this solution in the algorithms solving the first problem presented above.

This paper is organized as follows. The problem is stated and previous work on parameterizing projective structure and motion is reviewed in §2. Mapped coordinates are presented in §3 and applied to minimal structure parameterization. The new algorithms for minimally parameterizing the projective two-view motion are given in §§4, 5 and 6. Experiments conducted on both simulated data and real images are shown in §§7 and 8 respectively. Finally, our conclusions are drawn in §9.

2 Problem Statement and Previous Work

In this section, we give some mathematical notations and preliminaries. We recall the principle of bundle adjustment and formally state the problem of the coordinate frame ambiguity. We then review existing approaches for the general and the two-view cases before introducing our contributions. Finally, we state and review existing solutions to the problem of minimally parameterizing homogeneous entities.

2.1 Notations and Preliminaries

We use perspective projection to model cameras which are therefore represented as homogeneous 3×4-matrices, denoted as P^i, $i \in 1 \dots n$ for the n views considered. Each such matrix has 12 algebraic DOF (degrees of freedom) plus 1 homogeneity constraint which correspond to its 11 essential DOF. Each camera gives rise to a view, which represents its geometric properties, e.g. its position in *3D* projective space. The term "motion" will often be used to designate the set of views.

Points in *3D* projective space are modeled as homogeneous 4-vectors, denoted as $\mathbf{Q}^j$, $j \in 1 \dots m$ for the m points considered. Each such vector has 4 algebraic DOF plus 1 homogeneity constraint which correspond to its 3 essential DOF. The term "structure" will often be used to designate the set of points.

Actual image points are represented as homogeneous 3-vectors and denoted as $\mathbf{q}^{i,j}$. The image point $\mathbf{q}^{i,j}$ is defined provided that the j-th *3D* point projects onto the i-th view. To handle this easily in mathematical expressions, we define $w^{i,j}$ as:

$$w^{i,j} = \begin{cases} 1 & \text{if point } \mathbf{Q}^j \text{ is visible in view } \mathsf{P}^i; \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Parameterizing the Problem

Let us represent the parameterizations of the motion and structure by vectors $\boldsymbol{\pi}_{\mathsf{P}}$ and $\boldsymbol{\pi}_{\mathbf{Q}}$ respectively. From there, we can define the parameterization functions $\mathcal{P}_{\mathsf{P}}$ and $\mathcal{P}_{\mathbf{Q}}$ for the motion and structure respectively as:

$$\mathsf{P}^1, \ldots, \mathsf{P}^n \xrightarrow{\mathcal{P}_{\mathsf{P}}} \boldsymbol{\pi}_{\mathsf{P}} \text{ and } \mathbf{Q}^1, \ldots, \mathbf{Q}^m \xrightarrow{\mathcal{P}_{\mathbf{Q}}} \boldsymbol{\pi}_{\mathbf{Q}}.$$

If these functions are one-to-one then the parameterization is complete, i.e. it can represent all non intrinsically singular configurations and the functions ${\mathcal{P}_{\mathsf{P}}}^{-1}$ and ${\mathcal{P}_{\mathbf{Q}}}^{-1}$ exist, allowing bundle adjustment to be processed as indicated in the next paragraph.

Minimal parameterizations are characterized by the fact that their number of parameters, i.e. the dimensions of $\boldsymbol{\pi}_{\mathsf{P}}$ and $\boldsymbol{\pi}_{\mathbf{Q}}$, equals the number of essential DOF of the problem. This is formally stated in §2.4.

2.3 Bundle Adjustment

Using previous notations, a bundle adjustment process can be conducted by finding:

$$\underset{\boldsymbol{\pi}_{\mathsf{P}}, \boldsymbol{\pi}_{\mathbf{Q}}}{argmin}\, \mathcal{C}({\mathcal{P}_{\mathsf{P}}}^{-1}(\boldsymbol{\pi}_{\mathsf{P}}), {\mathcal{P}_{\mathbf{Q}}}^{-1}(\boldsymbol{\pi}_{\mathbf{Q}})),$$

where $\mathcal{C}$ is the *cost function* or the *error* defined as the sum of squared euclidean distances between actual and reprojected image points:

$$\mathcal{C}(\mathsf{P}^1, \ldots, \mathsf{P}^n, \mathbf{Q}^1, \ldots, \mathbf{Q}^x m) = \sum_{i=1}^{n} \sum_{j=1}^{m} w^{i,j}\ d^2(\mathbf{q}^{i,j}, \mathsf{P}^i \mathbf{Q}^j),$$

where $d(.,.)$ is the euclidean distance. It is important to note that $\mathcal{C}$ is invariant to *2D* rigid transformations, such as rotation and translation, but not to affine or projective changes, which are sometimes applied to image points to simplify the form of projection matrices, see §2.5.

Bundle Adjustment provides the *maximum likelihood estimate* when image points are corrupted by an i.i.d. centered gaussian noise. It is said to be *optimal* in this sense.

In this paper, bundle adjustments are conducted using the iterative non-linear optimization algorithm Levenberg-Marquardt [4] with numerical differentiation. Such an algorithm needs an initial estimate. In the case of projective structure and motion, there exist many algorithms that can provide a sub-optimal initial solution, see e.g. [20].

Another important aspect of the estimation process that has to be considered is the data conditioning. This can be enhanced by using the method described in [5] that consists in normalizing the data, i.e. center and scale image points such that their mean distance to the center of the image is $\sqrt{2}$. Obviously, the scales used for each image, and in the non-isotropic case, for each image axis, are different and have to enter the cost function so as to preserve the relative weight of each term.

2.4 The Coordinate Frame Ambiguity

An important issue to consider for bundle adjustment is which coordinate frame is used to express the recovered structure and motion. This coordinate frame ambiguity which constitutes gauge freedoms is an inherent problem to structure and motion recovery and may affect numerical stability and therefore convergence properties of the estimation process. Similarly, one has to keep in mind that an appropriate choice of coordinate frame can be fruitful for the estimation process.

In the metric case, this problem can be easily solved since a single view can be used to define the coordinate frame [18], up to a global scale factor, while in the projective case, these gauge freedoms constitute one of the main problems to be dealt with for bundle adjustment, since the algebraic structures necessary to manipulate for fixing the coordinate frame are much more complex. In particular, one view does not define a unique projective basis while two views provide too many constraints.

To understand the problem, let us consider a particular reconstruction $(\mathsf{P}, \mathbf{Q})$, expressed in a particular coordinate frame, and its error $\mathcal{C}(\mathsf{P}, \mathbf{Q})$. We drop the index for clarity of presentation. This error is estimated from both actual and reprojected image points. Let us express the reconstruction in another coordinate frame, linked to the current one by a *3D* homography H. Each projection matrix P and *3D* point $\mathbf{Q}$ transforms into respectively $\mathsf{P}\mathsf{H}^{-1}$ and $\mathsf{H}\mathbf{Q}$. It appears that this does not change the image coordinates of reprojected points:

$$\mathsf{P}\mathsf{H}^{-1}\mathsf{H}\mathbf{Q} \sim \mathsf{P}\mathbf{Q},$$

and therefore leaves the error, the value of the cost function, invariant:

$$\mathcal{C}(\mathsf{P}\mathsf{H}^{-1}, \mathsf{H}\mathbf{Q}) = \mathcal{C}(\mathsf{P}, \mathbf{Q}).$$

The coordinate frame is therefore ambiguous since it is not measured by the cost function.

A change of projective basis, i.e. the *3D* homography H, has 15 DOF. From this, we can state the number of essential DOF of the problem, which is equal to that of both structure and motion minus the coordinate frame ambiguity:

$$\#\text{DOF} = 12n + 4m - n - m - 15 = 11n + 3m - 15.$$

We can now characterize minimal parameterizations as $\dim \boldsymbol{\pi}_{\mathsf{P}} + \dim \boldsymbol{\pi}_{\mathbf{Q}} = \#\text{DOF}$. In the other case, the problem is said to be overparameterized.

2.5 Means to Parameterize

Several means have been used to handle these gauge freedoms. They can be classified as follows:

free gauges consist in overparameterizing the problem using all algebraic parameters, i.e. all entries of P and $\mathbf{Q}$ and letting the gauges evolve freely;

structure-based relies on fixing the coordinates of five reconstructed points. The five chosen points need to be in general configuration;

motion-based is achieved by using a specific form of camera matrices P;

image-based relies on using transformations on image points to e.g. give a simplified form to projection matrices and some *3D* points. This does not constrain the gauge freedoms but may be useful to achieve a minimal parameterization. For example, [9] proposes to choose the image and *3D* coordinates such that each camera matrix writes as $\mathsf{P}^i \sim (\mathrm{I}\ \mathbf{a}^i)$ where $\mathbf{a}^i$ are inhomogeneous 3-vectors that have fixed values for the five first cameras;

overparameterization and penalty consists, as in the free gauge case, in overparameterizing the problem and preventing the gauge to drift by adding soft constraints via hallucinated measurements. For example, [13] proposes a set of constraints on camera matrices that fix the coordinate frame and add penalty terms to the cost function to make the estimation process convergence to a solution which approximately satisfies these constraints.

Each means has advantages and drawbacks, in terms of numerical conditioning, number of parameters and number of hallucinated measurements added to the cost function. In more detail:

free gauges has the advantage of simplicity of implementation but might cause numerical instabilities and the estimator to diverge. It also involves estimating many more parameters than necessary. We test such a parameterization in §§7 and 8 under the name *Free*;

structure-based has the advantage that the number of parameters to estimate is usually minimal but the estimation behaviour strongly relies on the choice made for the *3D* points that form the projective basis. Moreover, points in general configuration should be a priori known or detected;

motion-based has the advantage that slightly overparameterized solutions are simple. We test the parameterization of [7] in §§7 and 8 under the name *Partially Free*;

image-based often yields simple expressions of camera matrices. However, non euclidean image transformations that are used most of the time do not leave the cost function invariant and make the bundle adjustment sub-optimal;

overparameterization and penalty gives simple implementations and prevent the gauges to drift. However, superfluous parameters and measurements are used which increases the computational cost of the process. We test the parameterization of [13] in §§7 and 8 under the name *Normalized*.

Most of the existing methods combine these possibilities to completely or partially fix the coordinate frame.

Let us give some detail about the existing approaches. In [3, 8, 9], the approach followed is mixed structure- and image-based since the coordinate frame is fixed by setting five point coordinates to match the standard projective basis. Additional image transformations yield

simple closed-form expressions for several camera matrices, two [3], two to four [8] or many [9]. These parameterizations are minimal but not well-adapted to bundle adjustment since changes in the image coordinates do not leave the cost function invariant, yielding a sub-optimal bundle adjustment result.

In [17], a relative affine structure is introduced. This representation is related to plane+parallax [10, 15, 16, 19, 21]. Within such a framework, the first 3×3 sub-block of camera matrices is a plane homography corresponding to a *3D* plane not necessarily observed by the cameras. For the first view, this sub-block can be set to the identity matrix. The last columns of camera matrices are then the epipoles of each view with respect to the first one and a null vector for the first view. This framework has the advange of the motion-based approach but is not minimal and there is not any existing work on its minimal parameterization.

In [13], the author completely parameterizes all algebraic entities and prevents the gauge to drift by adding soft constraints, i.e. hallucinated measurements to the cost function, on the entries of projection matrices. While the results given by this approach are satisfactory in terms of convergence properties, its computational cost is non negligible since the problem is overparameterized and several artificial measurements are added.

Another possibility is to consider two views to incorporate the gauge constraints into the paremeterization. In this case, the other views, and the structure are represented as general homogeneous entities. We investigate this possibility in the next paragraph.

2.6 Reducing the Problem to the Two-View Case

As mentioned in many works, the problem of defining the coordinate frame with a special form for two camera matrices is equivalent to parameterizing the fundamental matrix, see e.g. [7, 11, 23]. Indeed, the fundamental matrix can be expressed from the two camera matrices and conversely, the latter can be recovered from it while setting, completely or partially, the coordinate frame attached to the two views. Without loss of generality, let us select the two first views to define the coordinate frame. In this case, the other camera matrices do not have any special form and write as general homogeneous 3×4-matrices. In the general case, the more distant views, e.g. the first and last, of the image sequence constitute a good choice for the reference views since, if enougth corresponding points are avaible between them, they are less likely to fall in a singular case that close views.

From this point, as we only manipulate two views, we use the notations P and P' to designate the two corresponding camera matrices.

In [7], the author proposes to use the following camera matrices:

$$\mathsf{P} \sim (\mathrm{I}\ \mathbf{0}) \text{ and } \mathsf{P}' \sim (\mathsf{H}\ \mathbf{e}'), \tag{1}$$

where H is any plane homography and $\mathbf{e}'$ the second epipole. This representation, related to that of [10, 16, 17, 19, 21] is ambiguous since H can be related to any *3D* plane and the relative scale between H and $\mathbf{e}'$ is not defined. The problem is therefore overparameterized.

In [12], the authors propose to characterize a particular plane homography called the *canonic plane homography* and denoted as $\mathsf{H}^\star$. It is linked to the fundamental matrix by the following relation:

$$\mathsf{H}^\star \sim [\mathbf{e}']_\times \mathsf{F}, \tag{2}$$

and is the only plane homography satisfying the constraint:

$$\mathsf{H}^{\star\mathsf{T}} \mathbf{e}' = \mathbf{0}. \tag{3}$$

Under this constraint, and as it is an homogeneous matrix, $\mathsf{H}^\star$ has 9-1-3=5 essential DOF. Therefore, an epipole, i.e. 2 essential DOF and the canonic plane homography, i.e. 5 essential DOF are strictly equivalent to the epipolar geometry, i.e. 7 essential DOF. The fundamental matrix can be deduced from any plane homography and the second epipole as $\mathsf{F} \sim [\mathbf{e}']_\times \mathsf{H}$.

By choosing $\mathsf{H} \sim \mathsf{H}^\star$ for the projection matrices (1), the ambiguity on the choice of H is removed. It has also been proposed in [12] to use normalized versions of both $\mathsf{H}^\star$ and $\mathbf{e}'$, so as to remove the ambiguity on their relative scale. The *canonic projection matrices*, defining the *canonic reconstruction basis* are therefore given by:

$$\mathsf{P} \sim (\mathrm{I}\ \mathbf{0}) \text{ and } \mathsf{P}' \sim (\mathsf{H}^\star\ \mathbf{e}'), \text{ with } ||\mathsf{H}^\star||^2 = ||\mathbf{e}'||^2 = 1. \tag{4}$$

However, [12] does not propose any minimal parameterization of these entities. In [11], the fundamental matrix is minimally parameterized but the analysis is restricted to the case where the epipoles are finite.

In [23], the author extends the minimal parameterization of [11] to the general case by using maps. Indeed, the parameterization used depends on the two-view configuration. This is equivalent to parameterizing the rank-2-ness of the fundamental matrix. A closed-form expression of the fundamental matrix is attached to each of 36 maps, which makes the method quite complicated and error-prone to implement. Another solution has been proposed in [2] where the number of maps has been reduced to 3 using image transformations.

In the sequel, we do not make any distinction between parameterizing the canonic projection matrices or the fundamental matrix, since they are strictly equivalent. Parameterizing the canonic projection matrices will also be equivalent to parameterizing the second canonic projection matrix since the first one has a fixed form. We also consider a slightly different version of the original canonic projection matrices since we fix the relative scale between $\mathsf{H}^\star$ and $\mathbf{e}'$ by another means.

2.7 Our Contribution: Minimally Parameterizing the Projective Two-View Motion

In this paper, we state that techniques for estimating the fundamental matrix with a minimal number of parameters fall into three categories:

1. using maps [2, 23], i.e. different closed-form expressions of algebraic entities in terms of the parameters, to model all situations;

2. using a local update via decomposition;
3. using image transformations [2], to restrict the number of cases.

The first two categories give a way to estimate rank-2 3×3 matrices while the last one is specific to the fundamental matrix estimation. The second category is novel. In this paper, we propose three new algorithms to illustrate each category. In more detail, these algorithms consist in:

1. considering the canonic plane homography and parameterizing it while enforcing the constraint (3). This yields 7 parameters, as well as 2 map indices, that represent the two-view motion. These map indices, related to the mapped coordinates of §3 are necessary to recover the canonic plane homography from its 7 parameters. They are equivalent to the map index used in [2, 23].
2. factorizing the fundamental matrix using an SVD (singular value decomposition) and scaling it such that its highest singular value is 1. The other non null singular value and the two 3×3 rotation matrices obtained through the SVD can then be locally updated using 7 parameters;
3. using the fact that both the cost function considered for bundle adjustment and the data conditioning are invariant to image rotations around the image center to ensure a particular configuration of the epipoles. This leads to a unique and minimal parameterization of the fundamental matrix.

All these algorithms necessitate to minimally parameterize homogeneous entities. We state this problem in the next paragraph.

2.8 The Homogeneity Ambiguity

The second problem to be dealt with for projective bundle adjustment is that of the homogeneity of algebraic entities. Indeed, the algebraic representation of *3D* reconstructed points and camera matrices are defined up to scale, in other words, a $p+1$ parameters entity has only p essential DOF. This must be taken into account to prevent a bundle adjustment process to diverge and to save computation time by estimating only the minimal number of parameters, i.e. p.

A common strategy is to overparameterize these entities by using all their homogeneous parameters and hallucinated measurements to impose soft constraints on the norm of each entity. For example, a reconstructed point $\mathbf{Q} \sim (Q_1\ Q_2\ Q_3\ Q_4)^\mathsf{T}$ is parameterized using all its 4 elements and the constraint that its norm should be unity is imposed by using the artificial measurement $||\mathbf{Q}|| - 1 = 0$. The drawback of this method is that more parameters than necessary are estimated using more measurements than necessary, which increases the computational cost of the estimation process.

In this paper, we propose to use what we call *mapped coordinates* to perform optimization on the minimum number of parameters without any additional measurements.

3 Mapped Coordinates

Homogeneous algebraic entities have an inherent ambiguity or gauge freedom as they are only defined up to a non zero scale factor. Consequently, they are not minimal since $p+1$ coordinates represent p essential DOF.

We define a tool called mapped coordinates that locally removes the homogeneity, in other words produces a minimal version of an homogeneous entity. More precisely, the homogeneous entity is decomposed into an unhomogeneous one plus a map index which is necessary to subsequently recover the homogeneous entity. We show that this decomposition has the property that, when the coordinates undergo a small variation, only its inhomogeneous part is likely to vary, the map being constant. Consequently, this system is particularly well adapted for non-linear optimizers such as Levenberg-Marquardt [4], where the map can be re-estimated at each step of the optimization. A practical algorithm for using mapped coordinates is given in table 1.

Before going into further detail, let us give the definition of three terms that we use in the sequel:

- *map*: unique and bidirectional correspondence between a set of parameters and algebraic entities;
- *parameterization*: bidirectional correspondence between a set of parameters and algebraic entities. A parameterization can consist of several maps, the most appropriate one being selected according to the case considered;
- *map index*: number of the map used in a parameterization. This number is necessary for the unparameterization step.

Let us give some detail about mapped coordinates. We consider the case of homogeneous vectors of $\mathbb{P}^n$, which is not a restriction, the method being valid for homogeneous entities of any dimension. In more detail, a $(p+1)$-vector $\mathbf{v}$, can be decomposed into a p-vector $\widetilde{\mathbf{v}}$ and a map index $v \in \{1, \ldots, p+1\}$, the index of a coefficient to be fixed. This coefficient is chosen as the maximum one of $\mathbf{v}$. An important property is that slightly changing $\mathbf{v}$ does not, in general, affect v but only $\widetilde{\mathbf{v}}$, and if v is affected, it will usually not create numerical instability (in the sense that the maximum coefficient of $\mathbf{v}$ will not tend towards zero during e.g. optimization).

The second step of the mapping requires to choose an element, the v-th, that is non null, to divide the vector. The existence of such a non null element is ensured since an homogeneous entity can not be completely null. Let us assume that each coefficient has the same probability to vary at each optimization step. In this case, the element that is the most unlikely to become null after such a step is the one that has the maximum absolute value.

Minimal Structure Parameterization. In this paragraph, we show how to minimally parameterize the structure using mapped coordinates. In this case, the m *3D* reconstructed points have $4m$ algebraic parameters.

Let $\mathbf{v}$ be an homogeneous $(p+1)$-vector. Any other homogeneous entity (matrix, tensor) can be brought back to this case by stacking its elements into a single vector. The inhomogeneous p-vector $\widetilde{\mathbf{v}}$ represents the mapped coordinates of $\mathbf{v}$ whereas the integer v represents its map index. *Mapping* $(\mathbf{v} \rightarrow (\widetilde{\mathbf{v}}, v))$*:* • choose v such that $v = argmax_i \lvert v_i \rvert$; • $\widetilde{\mathbf{v}} = \frac{\mathbf{v}_{/v}}{v_v}$. *Unmapping* $((\widetilde{\mathbf{v}}, v) \rightarrow \mathbf{v})$*:* • $\mathbf{v} \sim \widetilde{\mathbf{v}}_{v \leftarrow 1}$.

Table 1: Mapped coordinates for homogeneous entities. Only $\widetilde{\mathbf{v}}$ has to be included in optimization processes. The notation $\mathbf{v}_{/v}$ designates the vector formed from all elements of $\mathbf{v}$ besides the v-th and $\mathbf{v}_{v \leftarrow \alpha}$ the vector $\mathbf{v}$ with value α inserted at the v-th position.

Let us decompose each point coordinate $\mathbf{Q}^j$ using its mapped coordinates $\widetilde{\mathbf{Q}}^j$ and its map index Q^j, as indicated in table 1. Each $\widetilde{\mathbf{Q}}^j$ has 3 parameters. We get therefore a minimal set of $3m$ parameters since the map indices do not have to be included into optimization processes.

4 Direct Minimal Parameterization of the Canonic Projection Matrices

In this section, we give a solution to the minimal parameterization of the two-view projective motion by considering the canonic projection matrices (4). Our algorithm yields a parameterization consisting of 7 parameters plus 2 maps. Unlike previous approaches based on maps [2, 23], the algorithm is very simple to use and does not require several closed-form expressions of the different algebraic entities.

The idea of the algorithm is to choose which parameters of $\mathsf{H}^\star$ are representative while enforcing the constraint (3).

For that purpose, let us examine this constraint in more detail. It can be rewritten as $\mathbf{h}_i^\mathsf{T} \mathbf{e}' = 0$, $i \in \{1 \dots 3\}$, where $\mathbf{h}_i$ represents the i-th column of $\mathsf{H}^\star$, i.e. 3 equations linear in the entries of $\mathsf{H}^\star$. If we assume $\mathbf{e}'$ to be known, we can eliminate one element of each $\mathbf{h}_i$, say the l-th, of the parameterization, since it can be recovered while enforcing the constraint as:

$$h_{l,i} = -\frac{\sum_{z \neq l} h_{z,i} e'_z}{e'_l}.$$

If the index l is the same for all the three columns of $\mathbf{h}_i$, the complete l-th row can be recovered from the other entries of $\mathsf{H}^\star$ and may be excluded from the parameterization.

Obviously, this necessitates $e'_l \neq 0$. As in the case of mapped coordinates, the element of $\mathbf{e}'$ that is the most unlikely to become zero under small changes in the coordinates is $argmax_i|e'_i|$, which is the map of $\mathbf{e}'$. Therefore, *we exclude the row of* $\mathsf{H}^\star$ *whose index corresponds to the map of* $\mathbf{e}'$ *from the parameterization.*

From there, 3 parameters of $\mathsf{H}^\star$ have been eliminated, the 6 remaining ones are still homogeneous and their 5 mapped coordinates are used. The 2 mapped coordinates of $\mathbf{e}'$ finally provide the minimal parameterization. The corresponding algorithm is given in table 2.

Let $\mathsf{H}^\star$ and $\mathbf{e}'$ be the canonic plane homography and the second epipole, i.e. the constraint ${\mathsf{H}^\star}^\mathsf{T}\mathbf{e}' = \mathbf{0}$ holds. The inhomogeneous 7-vector $\boldsymbol{\pi}$ represents their parameterization, and $\mathbf{m}$ a set of 2 map indices necessary to subsequently unparameterize $\boldsymbol{\pi}$ to recover $\mathsf{H}^\star$ and $\mathbf{e}'$. The parameterization routine has to be called before each iteration of the optimization process while the unparameterization has to be called before each evaluation of the cost function.
Parameterization $((\mathsf{H}^\star, \mathbf{e}') \rightarrow (\boldsymbol{\pi} + \mathbf{m}))$:

- Compute the *mapped coordinates of* $\mathbf{e}'$, $\widetilde{\mathbf{e}}'$ and its map index e';
- Extract $\bar{\mathsf{H}}$, the 2×3 sub-matrix obtained by removing the e'-th row of $\mathsf{H}^\star$;
- Compute the *mapped coordinates of* $\bar{\mathsf{H}}$, $\widetilde{\bar{\mathsf{H}}}$ and its map index $\bar{H}$;
- Build the *set of parameters* $\boldsymbol{\pi} = \{\widetilde{\mathbf{e}}', \widetilde{\bar{\mathsf{H}}}\}$, consisting of the mapped coordinates of $\mathbf{e}'$, i.e. 2 parameters, and that of $\widetilde{\bar{\mathsf{H}}}$, i.e. 5 parameters.
- Store the *map indices* as $\mathbf{m} = \{e', \bar{H}\}$.

Unparameterization $((\boldsymbol{\pi} + \mathbf{m}) \rightarrow (\mathsf{H}^\star, \mathbf{e}'))$:

- Recover $\mathbf{e}'$ using an *unmapping*;
- Recover $\bar{\mathsf{H}}$ using an *unmapping*;
- Compute the e'-th missing row and form $\mathsf{H}^\star$ from $\bar{\mathsf{H}}$ using $h_{e',i} = -\sum_{z\neq e'} h_{z,i}e'_i$.

Table 2: Minimal parameterization of the canonic projection matrices.

5 Factorization of the Fundamental Matrix

In this section, we propose to consider the SVD of the fundamental matrix as a means to minimally parameterize the two-view motion. Such a decomposition is composed of two 3×3 rotation matrices and two non null singular values. Each rotation matrix can be locally updated using 3 parameters, while only the ratio of the two singular values is relevant, due to the homogeneity of the fundamental matrix. This provides a 7-parameter update scheme of the decomposition. We can then rebuild the fundamental matrix or directly the canonic plane homography that composes the canonic projection matrices.

Let us denote the SVD of the fundamental matrix as $\mathsf{F} \sim \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$, where U and V are 3×3 orthogonal matrices and Σ a diagonal one containing the singular values of F. As F is a rank-2 matrix, $\Sigma \sim \mathrm{diag}(\lambda_1, \lambda_2, 0)$ where λ_1 and λ_2 are strictly positive scalars and $\lambda_1 > \lambda_2$. As F is an homogeneous entity, we can scale its SVD such that:

$$\mathsf{F} \sim \mathsf{U} \begin{pmatrix} 1 & & \\ & \lambda = \frac{\lambda_2}{\lambda_1} & \\ & & 0 \end{pmatrix} \mathsf{V}^\mathsf{T}.$$

Each rotation matrix is updated using 3 parameters with the Rodriguez formula [1] and λ is completely included into the parameterization.

From there, the fundamental matrix can be recovered as:

$$\mathsf{F} \sim \mathbf{u}_1\mathbf{v}_1^\mathsf{T} + \lambda\mathbf{u}_2\mathbf{v}_2^\mathsf{T},$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the columns of U and V respectively. The second epipole is the last column of U: $\mathbf{e}' \sim \mathbf{u}_3$. The canonic plane homography can be formulated using equation (2) as:

$$\begin{aligned} \mathsf{H}^\star &\sim [\mathbf{e}']_\times \mathsf{F} \\ &\sim [\mathbf{u}_3]_\times \left(\mathbf{u}_1\mathbf{v}_1^\mathsf{T} + \lambda\mathbf{u}_2\mathbf{v}_2^\mathsf{T}\right). \end{aligned}$$

Since U is an orthogonal matrix, $[\mathbf{u}_3]_\times\mathbf{u}_1 = \mathbf{u}_2$ and $[\mathbf{u}_3]_\times\mathbf{u}_2 = -\mathbf{u}_1$ which yields:

$$\mathsf{H}^\star \sim \mathbf{u}_2\mathbf{v}_1^\mathsf{T} - \lambda\mathbf{u}_1\mathbf{v}_2^\mathsf{T}.$$

The corresponding algorithm is given in table 2.

6 From the Epipolar Geometry to *2D* Entities

In this section, we consider the components of the epipolar geometry described in e.g. [11], i.e. the two epipoles and the epipolar transformation. We parameterize directly these entities using mapped coordinates. The tricky part of the approach is to establish the link with the fundamental matrix or equivalently the canonic plane homography. Such a work has been done in [2, 23] with considering multiple closed-form expressions for the fundamental matrix. In this paper, we avoid the use of these multiple expressions by using specific image transformations. These transformations are chosen such that they leave invariant the cost function of the bundle adjustment and they preserve data conditioning.

6.1 The Epipolar Geometry as Two Epipoles and the Epipolar Transformation - Parameterization

Let us recall the signification of the epipolar geometry. Following the notations of figure 1, [11] states that the epipolar geometry can be decomposed into the two epipoles $\mathbf{e}$ and $\mathbf{e}'$

Let F be the fundamental matrix. The inhomogeneous 7-vector $\boldsymbol{\pi}$ is used to step-by-step update the estimate. The parameterization needs to be called only once, before starting the optimization process while the unparameterization has to be called before each evaluation of the cost function.

Parameterization ($\mathsf{F} \rightarrow \boldsymbol{\pi}$):

- Compute $\mathsf{F} \sim \mathsf{U} \begin{pmatrix} 1 & & \\ & \lambda & \\ & & 0 \end{pmatrix} \mathsf{V}^\mathsf{T}$, a scaled SVD of the fundamental matrix;
- Set the *parameters* as $\boldsymbol{\pi} = \{\mathbf{u}, \mathbf{v}, \lambda\}$ where $\mathbf{u} = \mathbf{v} = \mathbf{0}_3$ are needed to update U and V respectively.

Unparameterization ($\boldsymbol{\pi} \rightarrow (\mathsf{H}^\star, \mathbf{e}')$):

- Update U: $\mathsf{U} \leftarrow \mathsf{U}(\mathrm{I} + \sin\theta_1[\mathbf{n}_1]_\times + (1 - \cos\theta_1)[\mathbf{n}_1]_\times^2)$ where $\theta_1 = ||\mathbf{u}||$ and $\mathbf{n}_1 = \frac{\mathbf{u}}{\theta_1}$;
- Update V: $\mathsf{V} \leftarrow \mathsf{V}(\mathrm{I} + \sin\theta_2[\mathbf{n}_2]_\times + (1 - \cos\theta_2)[\mathbf{n}_2]_\times^2)$ where $\theta_2 = ||\mathbf{v}||$ and $\mathbf{n}_2 = \frac{\mathbf{v}}{\theta_2}$;
- Recover the *canonic plane homography* as: $\mathsf{H}^\star \sim \mathbf{u}_2\mathbf{v}_1^\mathsf{T} - \lambda\mathbf{u}_1\mathbf{v}_2^\mathsf{T}$;
- Recover the *second epipole* as: $\mathbf{e}' \sim \mathbf{u}_3$.

Table 3: An SVD-based 7-parameter update scheme of the two-view motion.

and a *1D* homography between the two epipolar pencils called the epipolar transformation. In fact, we use a *1D* homography denoted as g between the points obtained by intersecting each epipolar pencil by a line that does not contain the epipole.

Each epipole is represented as an homogeneous 3-vector and the epipolar transformation as an homogeneous 2×2-matrix. Using mapped coordinates of these entities, we obtain 7 parameters to include into the optimization process.

The difficulty is to recover corresponding *2D* entities from this parameterization.

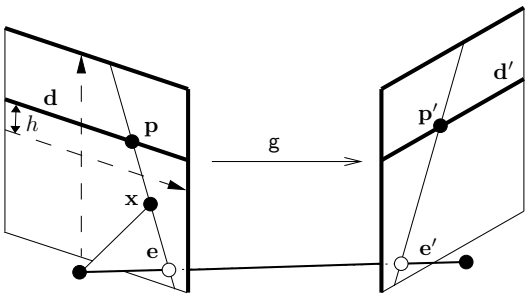


Figure 1: The epipolar geometry is composed of two epipoles and the epipolar transformation.

6.2 Recovering *2D* Entities - the Fundamental Matrix and Plane Homographies

In [11, 23], the following method is proposed. If we intersect the epipolar pencils by a line in each image, respectively $\mathbf{d}$ and $\mathbf{d}'$, we get one intersection point for each epipolar line, denoted $\mathbf{p}$ and $\mathbf{p}'$. These intersection points are related by a *1D* homography g [11], or, equivalently, a *2D* homography G with 3 DOF called the extended epipolar transformation [2].

The fundamental matrix can then be deduced as:

$$\mathsf{F} \sim [\mathbf{e}']_\times \mathsf{G} [\mathbf{d}]_\times [\mathbf{e}]_\times . \tag{5}$$

It is also shown in [2] that $\mathsf{G}[\mathbf{d}]_\times[\mathbf{e}]_\times$ is a plane homography that can be used to form the second projection matrix from equation (1).

The difficulty is to express the extended epipolar transformation G since its form depends on the intersection lines $\mathbf{d}$ and $\mathbf{d}'$ which must be chosen so that they do not contain the epipoles.

If we can ensure by some means that the epipoles do not lie on lines $\mathbf{d}$ and $\mathbf{d}'$ a priori known, we can derive a unique closed-form expression for the extended epipolar transformation G and therefore for the fundamental matrix.

In this paper, we propose to use image transformations to ensure that the epipoles do not lie on these lines.

6.3 Choosing the Intersection Lines and Taking Away the Epipoles

Let us consider the first view only, since the reasoning is exactly the same for the second one. We first examine the class of image transformations that are possible to use and then choose the form of the intersection line. We then give a means to estimate the best transformation that ensures that the epipole does not lie on the intersection line.

The constraints that we have when transforming the image data are the following:

- we want to leave the *cost function invariant*, so as to preserve the optimal character of bundle adjustment;
- we do not want to affect the *numerical conditioning* of the data, so as to preserve the convergence properties given by well-conditioned data.

The first condition imposes that the transformation used, denoted as T, must be a rigid *2D* transformation, i.e. rotation and translation. The second condition imposes that this transformation does not affect the spread of the data, i.e. does not change the coordinate of the center of gravity of image points. Consequently, the only possibility for T is a rotation around the image center, and the only DOF to choose is the angle of rotation.

Before we show how to best estimate this angle of rotation, let us fix the form of the intersection line. Obviously, $\mathbf{d}$ can not be the line at infinity, since if the epipole lies at infinity, an image rotation will keep it at infinity and so on $\mathbf{d}$. Also, $\mathbf{d}$ can not pass through

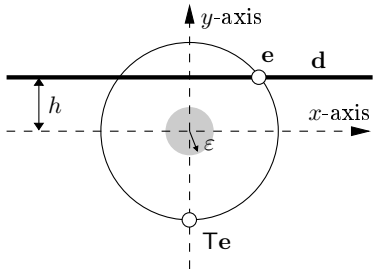


Figure 2: The epipole $\mathbf{e}$ is taken away from the a priori chosen intersection line $\mathbf{d}$ using the transformation T, a rotation around the image center.

the center of the image, since if $\mathbf{e}$ lies at the center, it will not be removed from $\mathbf{d}$ by the authorized rotation. The intersection line can therefore be any line different from the line at infinity and that does not pass through the image center. In this paper, we choose $\mathbf{d}$ as being parallel to the x-axis of the image:

$$\mathbf{d} \sim (0\ 1\ -h)^{\mathsf{T}}, \tag{6}$$

where $h \neq 0$ is the signed distance between $\mathbf{d}$ and the image center. Such a choice will simplify further expressions, notably thus of the extended epipolar transformation.

Let us show how to compute the transformation T as a pure centered rotation which moves the epipole farthest away from the intersection line. The farthest point from $\mathbf{d}$ after a rotation is situated on the y-axis, as illustrated on figure 2. The epipole after the optimal rotation writes then $\mathsf{T}\mathbf{e} \sim (0\ -k\ e_3)$ where $k = \sqrt{e_1^2 + e_2^2} = e_3 d(\mathbf{e}, \mathbf{c})$, where $\mathbf{c} \sim (0\ 0\ 1)^{\mathsf{T}}$ is the image center. Note that these formulas are valid even in the case where the epipole lie at infinity. The rotation that has to be applied to the image is then given by:

$$\mathsf{T} = \frac{1}{k} \begin{pmatrix} -e_2 & e_1 & 0 \\ -e_1 & -e_2 & 0 \\ 0 & 0 & k \end{pmatrix}.$$

The only problem is when the epipole lies too close to the image center, which causes k to tend towards zero and T becomes impossible to build. In this case, the angle of rotation α can not and need not to be estimated. Therefore, we simply detect this case and set $\alpha = 0$, i.e. the identity matrix as rotation. In practice, we use a threshold ε of the order of one pixel on the distance between the epipole and the image center.

The algorithm to compute T from $\mathbf{e}$ is given in table 4.

6.4 Deriving the Extended Epipolar Transformation

For the chosen intersection line, the intersection point writes $\mathbf{p} \sim (p_1\ hp_2\ p_2)^{\mathsf{T}}$. Obviously, for a fixed h, only two entries are independent, the first one and the second or the third

Let $\mathbf{e} \sim (e_1\ e_2\ e_3)^\mathsf{T}$ be a given epipole at a given step of the estimation process, T is the transformation to apply to image data to make the parameterization as non singular as possible while preserving data conditioning and the error, i.e. the value of the cost function. The threshold ε is chosen to be of the order of one pixel.

- Let $k = \sqrt{e_1^2 + e_2^2}$;
- If $k < e_3\varepsilon$ then set $\mathsf{T} = \frac{1}{k}\begin{pmatrix} e_2 & -e_1 & 0 \\ -e_1 & -e_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ else set $\mathsf{T} = \mathrm{I}_3$.

Table 4: Given an epipole, this algorithm computes the centered image rotation such that this epipole comes the farthest as possible from the intersection line.

one. Let us represent $\mathbf{p}$ by its two first entries: $\bar{\mathbf{p}} \sim (p_1\ hp_2)^\mathsf{T}$. The epipolar transformation applies as:

$$\bar{\mathbf{p}}' \sim \mathsf{g}\bar{\mathbf{p}} \text{ with } \mathsf{g} \sim \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

and one can easily verify that the extented epipolar transformation can be recovered as:

$$\mathsf{G} \sim \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ c/h & 0 & d \end{pmatrix} \text{ and } \mathbf{p}' \sim \mathsf{G}\mathbf{p}. \tag{7}$$

The final algorithm is given in table 5.

7 Experiments on Simulated Data

In this section, we compare our methods to existing ones using simulated data. The test bench consists of 50 points lying inside a sphere with a radius of 1 meter observed by two cameras with a focal length of 1000 (expressed in number of pixels). Each of these cameras is looking at the center of the sphere and is situated at a distance of 10 meters from it. The baseline between the two cameras is 1 meter.

Points are generated in *3D* space, projected onto the images and corrupted by an additive centered gaussian noise with varying standard deviation.

We measure the two quantities depicted as characteristic of a bundle adjustment process, computational cost, i.e. CPU time to convergence and the error at convergence, versus the standard deviation of added image noise. We also measure the error of the current estimation as a function of time through the optimization processes. The plots correspond to median values over 100 trials. The bundle adjustments are initialized by the values obtained using the 8 point algorithm [5] and the triangulation method described in [6]. We compare the following algorithms:

Let F be the fundamental matrix. The inhomogeneous 7-vector $\boldsymbol{\pi}$ contains the parameterization of the epipolar geometry and $\mathbf{m}$ the map indices, which are used to subsequently recover the canonic projection matrices. The parameterization routine has to be called at each iteration of the optimization process, but only from step 2. since the epipoles are already known while the unparameterization has to be called before each evaluation of the cost function.
Parameterization ($\mathsf{F} \rightarrow (\boldsymbol{\pi}, \mathbf{m})$):

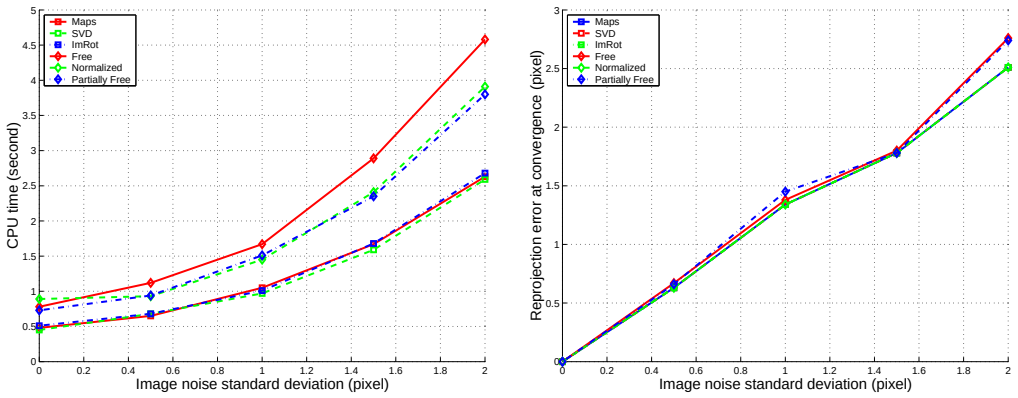
- Compute the *epipoles* and their *mapped coordinates* via an SVD of the fundamental matrix;
- Estimate and apply the *image transformations* T and T' to image points, the epipoles and the fundamental matrix;
- Compute the entries of the *epipolar transformation* g and its mapped coordinates from the linear system given by equation (5) and the form of G (7) and $\mathbf{d}$ (6);
- Compute the *mapped coordinates* $\widetilde{\mathsf{g}}$ of the *epipolar transformation* g and its map g;
- Set the *parameters* $\boldsymbol{\pi} = \{\widetilde{\mathbf{e}}, \widetilde{\mathbf{e}}', \widetilde{\mathsf{g}}\}$;
- Store the map indices as $\mathbf{m} = \{e, e', g\}$.

Unparameterization ($(\boldsymbol{\pi}, \mathbf{m}) \rightarrow (\mathsf{H}^\star, \mathbf{e}')$):

- Recover the *epipoles* and the *epipolar transformation* using unmappings;
- Form the *extended epipolar transformation* using equation (7);
- Form the *fundamental matrix* using equation (5);
- Form the *canonic plane homography* using equation (2).

Table 5: An approach based on parameterizing the epipolar geometry to subsequently recover the canonic projection matrices.

- *Maps*, *SVD* and *ImRot*: use the parameterizations given in §§4, 5 and 6 respectively;
- *Free*: uses an overparameterization with free gauges;
- *Normalized*: uses an overparameterization plus hallucinated measurements to prevent the gauge to drift [13];
- *Partially Free*: uses a partially free gauge by completely parameterizing the second camera matrix as described in [7];



(a) (b)

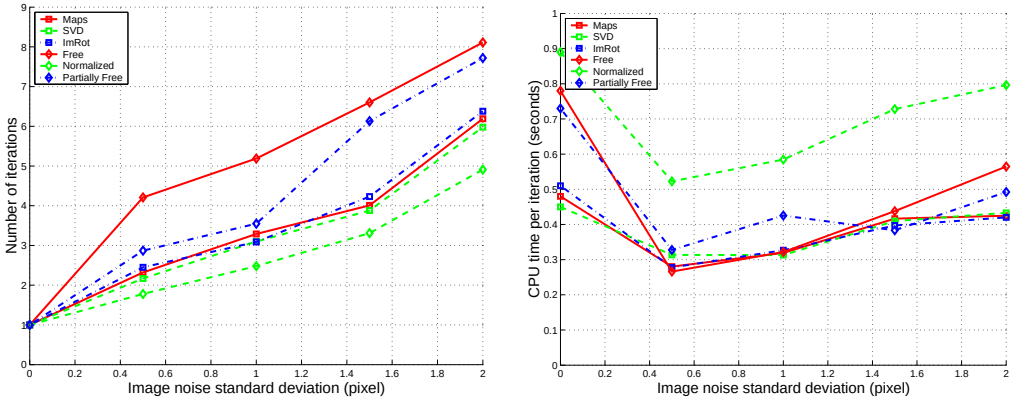
Figure 3: Comparison of (a): the CPU time to convergence and (b): value of the error at convergence versus varying image noise for different methods.

We observe on figure 3b that, roughly speaking, all methods converge to the same minimum of the cost function. Methods that have a slightly less reliable convergence than the others are *Free* and *Partially Free*.

Figure 3a shows that, for roughly the same convergence properties, there are quite big disparities between the computational cost of each method. The method that has the highest, i.e. the worst computational cost is *Free*, followed by *Normalized* and *Partially Free*. Finally, methods using the minimal number of parameters presented in this paper, *Maps*, *SVD* and *ImRot* have the lowest computational cost, roughly the same.

It is clear from these results which method one has to choose, for both simplicity, computational cost and convergence properties. However, in order to understand and explain the behaviour of the different methods, we have measured the number of iterations and the computational cost of these iterations. These results are shown on figure 4.

In more detail, we have found that methods *Free* or *Partially Free*, leaving the gauge drift freely have very bad convergence properties, performing more iterations, roughly the double, than the others, see figure 4a. Method *Normalized* performs a number of iterations smaller than all the other methods but involves solving a much more costly linear system at



(a) (b)

Figure 4: Comparison of (a): the number of iterations to convergence and (b): the computational cost of each iteration.

each iteration, see figure 4b. Methods presented in this paper, using the minimal number of parameters are trade-offs between the number of iterations and their computational cost: each iteration has a low computational cost and the number of iterations needed is in-between those of free gauge methods and the normalized method. This explains why these methods achieve the lowest total computational cost.

Finally, figure 5 shows the evolution of reprojection error for the different optimization processes. This experiment is useful in the sense that the time to convergence previously measured is highly dependent on how convergence is achieved, e.g. by thresholding two consecutive errors, and does not account for the ability of the algorithms to quickly, or not, reach almost the value of convergence. This experiment has been conducted using the same test bench as previously with a noise level on image point positions of 0.5 pixel. We can see on this figure that methods based on a minimal parameterization reach this value of convergence before the others. The *Normalized* and *Partially Free* methods take roughly twice the time to attain this value, while the *Free* method takes three times more.

8 Experiments on Real Images

In this section, we validate our algorithms using real images. We first consider the case of two images. In order to cover all possibilities for the epipoles to be close to the images or at infinity, we use pairs of the images taken from camera positions shown on figure 6. Initial values for structure and motion are computed as in the case of simulated data.

Results are shown in table 6. For each combination of image pair and each algorithm, we estimate the CPU time to convergence $\mathcal{T}$ and the error at convergence $\mathcal{E}$. These results

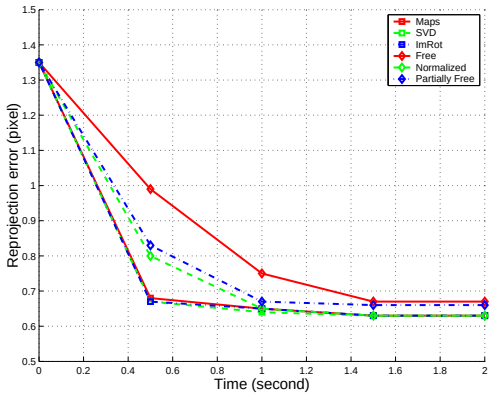


Figure 5: Comparison of the evolution of reprojection errors.

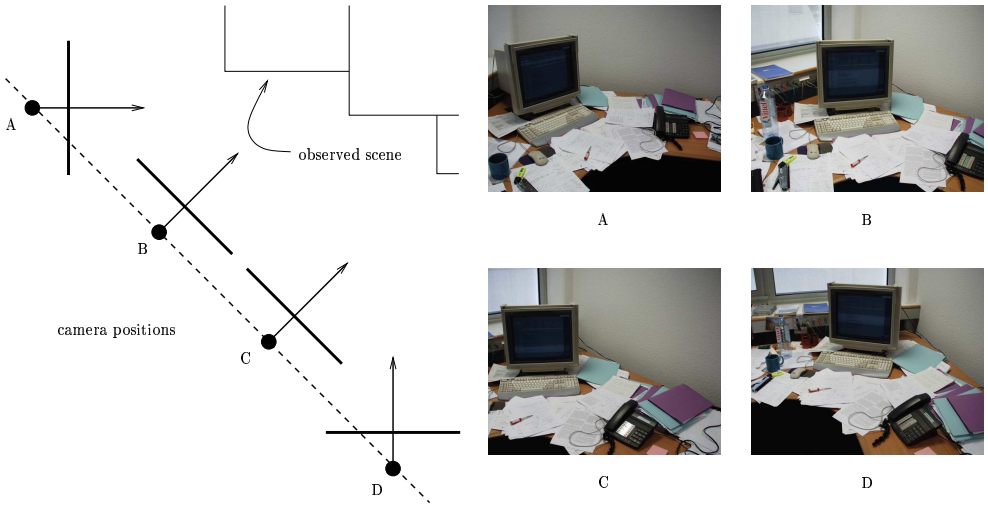


Figure 6: Real images used to validate the algorithms.

epipoles		views	Maps		SVD		ImRot		Free		Normalized		Partially Free	
e	e′		$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$
∞	∞	A, B	0.63	2.45	0.63	2.39	0.63	2.47	0.68	3.98	0.63	2.99	0.68	3.02
		A, C	0.71	2.38	0.71	2.41	0.71	2.40	0.77	4.01	0.71	3.56	0.71	3.71
∞	∞	A, D	0.45	2.03	0.45	1.76	0.45	2.19	0.57	3.13	0.45	3.09	0.45	2.93
∞	∞	B, C	0.88	3.53	0.88	3.39	0.88	3.55	1.23	6.70	0.88	5.12	0.88	4.63
∞	∞	B, D	0.59	2.33	0.59	2.10	0.59	2.81	0.59	3.99	0.59	3.41	0.59	3.56
		C, B	0.51	1.91	0.51	1.92	0.51	2.02	0.51	3.39	0.51	2.79	0.51	3.04
average $\bar{\mathcal{E}}$ and $\bar{\mathcal{T}}$			0.628	2.430	0.628	2.328	0.628	2.573	0.725	4.200	0.628	3.493	0.637	3.482
$1/(\bar{\mathcal{E}}\bar{\mathcal{T}})$			0.655		0.684		0.618		0.328		0.456		0.451	

Table 6: Error at convergence $\mathcal{E}$ and time to convergence $\mathcal{T}$ obtained when combining pairs of images from figure 6.

confirmed those obtained using simulated data. To make this clear, let us consider the last two rows of the table that show respectively the mean values $\bar{\mathcal{T}}$ and $\bar{\mathcal{E}}$ of $\mathcal{T}$ and $\mathcal{E}$ for each algorithm over the set of image pairs and the inverse of their product. This last measurement is a kind of score of the algorithms with respect to the tested set of images. This allows to rank the algorithms, according to both their convergence properties and computational costs: *SVD*, *Maps*, *ImRot*, *Normalized*, *Partially Free* and *Free*.

We have also tested the algorithms on all four images of figure 6. Initial values have been obtained by registering each view to a structure and motion obtained from the two first ones. The camera matrices of the two other views are modeled as general homogeneous 3×4 matrices and parameterized by 11 parameters using mapped coordinates. The results are the following: all algorithms converge with a final error of 0.73 pixels and their relative performances in terms of computation times to convergence were equivalent to those obtained in the case of two views.

9 Conclusions

We have studied the problem of minimally parameterizing the two-view motion, which is equivalent to constraining the coordinate frame in which the recovered structure and motion are expressed, using a special form of two camera matrices. We have proposed to classify the possible ways to achieve this in three categories and presented three new algorithms.

These new algorithms are simple to implement compared to the existing ones and experimental results show that they perform better, in terms of computational cost while achieving equivalent results in terms of convergence properties. Existing algorithms that do not constrain the gauge by any means perform worse than the others.

The algorithm that we recommend to use is the one called *SVD* since it is the most stable, simple to implement and it yields an easy closed-form for the Jacobian of the motion parameters that can be used in the Levenberg-Marquardt algorithm.

References

[1] N. Ayache. *Vision stéréoscopique et perception multisensorielle: application à la robotique mobile.* Science Informatique. InterEditions, 1989.

[2] A. Bartoli, P. Sturm, and R. Horaud. Projective structure and motion from two views of a piecewise planar scene. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 593–598, July 2001.

[3] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.

[4] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, 1981.

[5] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1064–1070, June 1995.

[6] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[7] R.I. Hartley. Projective reconstruction and invariants from multiple images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, October 1994.

[8] A. Heyden. Reduced multilinear constraints - theory and experiments. *International Journal of Computer Vision*, 1(30):5–26, 1998.

[9] A. Heyden and K. Åström. Algebraic varieties in multiple view geometry. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, 1996.

[10] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: a parallax based approach. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*, pages 685–688, 1994.

[11] Q.T. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.

[12] Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.

[13] P. F. McLauchlan. Gauge invariance in projective 3D reconstruction. In *Proceedings of the Multi-View Workshop, Fort Collins, Colorado, USA*, 1999.

[14] J. Oliensis. The error surface for structure and motion. Technical report, NEC, 2001.

[15] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 42–49, July 2001.

[16] H.S. Sawhney. 3D geometry from planar parallax. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA*, pages 929–934, 1994.

[17] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, 1996.

[18] C.C. Slama, editor. *Manual of Photogrammetry, Fourth Edition.* American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[19] G. Sparr. A common framework for kinematic depth, reconstruction and motion for deformable objects. In *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 471–482, May 1994.

[20] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 709–720. Springer-Verlag, April 1996.

[21] B. Triggs. Plane + parallax, tensors and factorization. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 522–538. Springer-Verlag, 2000.

[22] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle ajustment — a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

[23] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399